Amendments to the Claims:

Re-write the claims as set forth below. This listing of claims will replace all prior versions and listings, of claims in the application:

Listing of the Claims:

1. (currently amended) A method for superword register value numbering, the method comprising:

for an instruction having an operation code and value numbers of a plurality of sources:

hashing [[an]]the operation code and the value numbers of [[a]]the plurality of sources to generate a first hash value;

retrieving an operation value number from a first hash table based on the first hash value;

generating a result value number based on a previous ~~hit hash~~ value number and the operation value number; and

determining if the instruction is redundant by searching a second hash table using the result value number.

2. (currently amended) The method of claim 1 further comprising:

~~if~~ when the result value number is found within the second hash table, retrieving an output of the instruction from the second hash table.

3. (currently amended) The method of claim 1 further comprising:

[[if]]when the result value number is not found within the second hash table, writing the ~~operation~~ result value number to the second hash table.

4. (currently amended) The method of claim 1 further comprising:

prior to generating a result value number, retrieving the previous ~~bit hash value~~ value number.

5. (original) The method of claim 1 further comprising:

prior to retrieving the operation value number, comparing the first hash value with a first hash table.

6. (currently amended) The method of ~~claim 1~~claim 5 further comprising:

[[if]]when the first hash value is not within the first hash table, assigning the first hash value a multiple component hash value.

7. (original) The method of claim 1 wherein the operation value number is an n-tuple number.

8. (currently amended) The method of claim 1 wherein;

the instruction further includes a write mask; and

~~the step of~~ generating the result value number ~~includes:~~ includes, for each component in the write mask;

~~for each component in a write mask:~~

if the write mask value is [[true]]false, setting the result value number equal to the operation value number; and

if the write mask value is ~~false~~true, setting the result value number equal to [[a]]the previous value number.

9. (currently amended) The method of claim 1 wherein ~~the operation code and the value numbers are disposed within an instruction.~~ the instruction further including a previous bit and a write mask.

10. (currently amended) An apparatus for superword register value numbering, the apparatus comprising:

at least one memory device storing a plurality of executable instructions; and

at least one processor operably coupled to the at least one memory device, operative to receive the plurality of executable instructions such that the processor, in response to the executable instructions and for an instruction having an operation code and value numbers of a plurality of sources:

hashes [[an]]the operation code and the value numbers of [[a]]the plurality of sources to generate a first hash value;

retrieves an operation value number from a first hash table based on the first hash value;

generates a result value number based on a previous ~~bit hash~~ value number and the operation value number; and

determines if the instruction is redundant by ~~searches~~ searching a second hash table using the result value number.

11. (currently amended) The apparatus of claim 10 wherein the at least one processor further in response to the executable instructions:

[[if]]when the result value number is found within the second hash table, retrieves an output of the instruction from the second hash table.

12. (currently amended) The apparatus of claim 10 wherein the at least one processor further in response to the executable instructions:

[[if]]when the result value number is not found within the second hash table, writes the ~~operation result~~ value number to the second hash table.

13. (currently amended) The apparatus of claim 10 wherein the at least one processor further in response to the executable instructions:

prior to generating a result value number, retrieves the previous ~~bit hash~~ value number.

14. (original) The apparatus of claim 10 wherein the at least one processor further in response to the executable instructions:

prior to retrieving the operation value number, compares the first hash value with a first hash table.

15. (currently amended) The apparatus of ~~claim 10~~claim 14 wherein the at least one processor further in response to the executable instructions:

[[if]]when the first hash value is not within the first hash table, assigns the first hash value a multiple component hash value.

16. (currently amended) The apparatus of claim 10 wherein:

the instruction further includes a write mask; and

~~wherein~~ when the at least one processor generates the result value number, the at least one processor further in response to the executable ~~instructions:~~ and for each component in a write mask:

~~for each component in a write mask:~~

~~if the write mask value is true,~~ sets the result value number equal to the operation value number if the write mask value is false; and

~~if the write mask value is false,~~ sets the result value number equal to [[a]]the previous value number if the write mask value is true.

17. (original) The apparatus of claim 10 further comprising:

a superword register operably coupled to the processor, the superword register operative to store a plurality of instructions therein.

18. (original) The apparatus of claim 10 further comprising:

at least one hash memory device operably coupled to the at least one processor such that the at least one hash memory device is operative to store the first hash table and the second hash table.

19. (currently amended) A method for superword register value numbering, the method comprising:

for an instruction having an operation code and value numbers of a plurality of sources:

hashing [[an]]the operation code and the value numbers of [[a]]the plurality of sources to generate a first hash value;

comparing the first hash value with a first hash table;

retrieving an operation value number from the first hash table;

retrieving [[the]]a previous ~~bit hash~~ value number;

generating a result value number based on the previous ~~bit hash~~ value number and the operation value number;

searching a second hash table using the result value number;

if the result value number is found within the second hash table, retrieving an output of the instruction from the second hash table; and

if the result value number is not found within the second hash table, writing the ~~operation~~ result value number to the second hash table.

20. (currently amended) The method of claim 19 further comprising:

[[if]]when the first hash value is not within the first hash table, assigning the first hash value a multiple component hash value.

21. (currently amended) The method of claim 19 wherein;

the instruction further includes a write mask; and

~~the step of~~ generating the result value number ~~includes~~ includes, for each component in the write mask,

~~for each component in a write mask:~~

if the write mask value is [[true]]false, setting the result value number equal to the operation value number; and

if the write mask value is ~~false~~true, setting the result value number equal to [[a]]the previous value number.